REMARKS

In the Final Office Action mailed August 24, 2006, the rejections of claims 1-24 pending in the present application have been maintained by the Examiner. Reconsideration of the Examiner's rejections of claims 1-24 is respectfully requested in view of the reasons set forth herein.

In the Final Office Action, claims 1-3, 7-11, 15-19 and 23-24 were rejected under 35 U.S.C. § 102(b) as allegedly being anticipated by ***Draves*** (U.S. Patent No. 5,802,590). The Applicants respectfully traverse the Examiner's rejections.

It is respectfully submitted that the Examiner erred in rejecting independent claims 1, 9, and 17. An anticipating reference, by definition, must disclose every limitation of the rejected claim in the same relationship to one another, as set forth in the claim. Independent claims 1, 9, and 17 set forth, among other things, requesting to execute at least one of the plurality of instructions or set of instructions by the software code running on the processor. Claims 1, 9, and 17 also set forth obtaining a second security ID associated with the software code and executing the requested instruction or set of instructions providing that the second security ID matches the first security ID.

The Examiner relies upon the ***Draves*** reference, asserting that all the elements of claims 1, 9 and 17 are taught by ***Draves***. The Applicants respectfully disagree.

Draves describes techniques for granting only authorized processes a secure access to a shared computer system resource. The system described by ***Draves*** ensures that a computer program is authorized to access a computer system resource. In particular, ***Draves*** refers to a process each concurrently executing computer program. See ***Draves***, column 1, lines 14-15. While each concurrently executing computer program is referred to as a process, various

resources include the central processing unit, main memory, and peripheral devices (e.g., disk drives and printers). See ***Draves,*** column 1, lines 14-19. Since processes frequently need to share resources, to help manage the various resources, a kernel maintains a resource table for each process. See ***Draves,*** column 1, lines 23 and lines 42-43. The kernel allows a process access to a resource only when the passed key matches the key for the resource that is stored in the resource entry. See ***Draves***, column 1, lines 16-20.

The Examiner, on page 5 of the Final Office Action, alleges that the process corresponds to "instructions", as set forth in claims 1, 9 and 17. See ***Draves*** column 3, lines 60-62. The Examiner further alleges a resource identifier comprising the handle/key pair associated with the same process /program/code corresponds to the second security ID associated with the software code. Applicants disagree. It is respectfully submitted that the Examiner's rejection is improper because the Examiner cannot satisfy two claim rejection features with the same element. That is, the Examiner uses the same "process" in ***Draves*** to satisfy the requirement of "instruction" and another distinct requirement of the "software code" that executes the "process," *i.e.*, instruction(s), as set forth in claim 1. The Examiner effectively ignores the teachings of ***Draves*** and the Applicants' Specification. This is clearly improper because it is in direct contravention to the Federal Circuit precedent expressed in Phillips v. AWH, Corp., 415 F.3d 1303 (Fed. Cir. 2005) (*en banc*). Based on the above-indicated standard, it is respectfully submitted that ***Draves*** fails to specifically point out the entirety of all the features set forth in claims 1, 9 and 17. Thus, it is respectfully submitted that each of these independent claims is allowable over the art of record for at least the aforementioned reasons.

At page 3 of the Final Office Action, the Examiner states that ***Draves*** on column 2, lines 27-31, discloses a system for ensuring that a computer program is authorized to access a

computer system resource. As noted by the Examiner, on page 4 of the Office Action dated April 19, 2006, on column 3, lines 39-41, the main feature of the ***Draves'*** invention is to provide a secure access to resources. The Examiner further notes that the system generates a system-wide resource table that has a resource entry for each allocated resource. Since each resource entry contains a preferably non-forgeable key that uniquely identifies the resource, the Examiner further indicates that ***Draves*** on column 3, lines 42-48, discloses that a kernel maintains a system-wide resource table that is a hash table and that contains a resource entry corresponding to each resource allocated by the kernel.

However, as can be seen from above, ***Draves*** at least does not teach requesting to execute at least one of the plurality of instructions or set of instructions by a software code running on the processor. Additionally, ***Draves*** does not teach obtaining a second security ID associated with the software code or comparing such a second security ID with a first security ID of the instructions. Instead of providing a specific citation(s) in the ***Draves*** reference relied upon to make the section 102 rejections, in the Final Office Action, at page 6, the Examiner simply makes a conclusory statement that the second security ID could be provided to a program which is attempting forgery, however, it would not be able to access the requested resources since its security ID/identifier/Key pair would not be the same with the first Security ID which is provided to some other program. In the Final Office Action, at page 6, the Examiner merely appears to suggest that the application programs described by ***Draves*** on column 23-25 such as word programs and spreadsheet program could have a shared memory but one of the program would be able to access the resource of the other program if and only if it has one and the same key pairs/identifier. According to the Examiner, access to the resource of the other program would be otherwise denied as described on column 3, lines 60-column 4, and line 11 in ***Draves***.

Since the Examiner fails to provide support for the rejections within the cited reference itself, the Examiner's section 102 rejection of independent claim 1 is flawed. Thus, notwithstanding the Examiner's allegations and suggestions as to the ***Draves*** system, the ***Draves*** reference fails to teach or suggest each claimed feature set forth in claim 1.

According to the Examiner and as understood by the Applicants, ***Draves*** discloses that when a process wishes to access the allocated resource, it passes the handle.backslash.key pair to the kernel. The kernel then examines the resource entry indexed by the passed handle to determine whether the passed key is equal to the key in the indexed resource entry. The Examiner points out that in ***Draves*** when no such resource entry is found, the kernel denies the process access to the resource. On the other hand, when a resource entry that contains a matching key is found, the kernel allows the process to access the resource." See ***Draves,*** column 4, lines 7-10. In this manner, the Examiner asserts that since ***Draves*** on column 3, lines 39-41 discloses a technique for providing processes a secure access to resources, it ensures that a computer program is authorized to access a particular resource and therefore it teaches all the features recited in claim 1. ***Draves*** does not, however, teach or suggest requesting to execute at least one of the plurality of instructions or set of instructions by the software code running on the processor, obtaining a second security ID associated with the software code and executing the requested instruction or set of instructions providing that the second security ID matches the first security ID.

For at least the aforementioned reasons, Applicants respectfully submit that the present invention is not anticipated by ***Draves*** and request that the Examiner's rejections of claim 1 and its dependent claims under 35 U.S.C. § 102(b) be withdrawn.

With regard to the rejections of independent claims 1, 9 and 17, the Examiner further asserts that ***Draves*** discloses an apparatus, comprising a processor for running code thereon, at column 3, lines 39-42 and column 1, lines 11-22 and Figure 2. In ***Draves***, the Examiner alleges that a kernel of an operating system inherently operates in the processor for providing secure access. On column 1, lines 39-42, in ***Draves*** the portion of the operating system that is responsible for the allocation and de-allocation of resources is referred to as the kernel. The kernel interacts with a shell and other programs as well as with the hardware devices on the system. The Examiner alleges that ***Draves*** further discloses, on column 3, lines 60-62, each process that is being defined as concurrently executing computer programs on column 1, lines 14-15. The Examiner states that each process of ***Draves*** corresponds to each of a plurality of instructions or a set of instructions set forth in claim 1. Furthermore, the Examiner states that ***Draves*** on column 3, lines 43-50 discloses that the kernel maintains a system-wide resource table that is a hash table and it contains a resource entry corresponding to each resource allocated by the kernel. Accordingly, the Examiner concludes that each and every limitation of the independent claims 1, 9, and 17 is disclosed by the ***Draves*** reference.

Instead of requesting to execute at least one instruction by the software code running on the processor and executing the requested instruction, in ***Draves***, the server process 302 sends a resource allocation request to the kernel 304 for sharing the resource with the client process 314. The handle/key pairs for the shared resource are passed by the server process 302. In ***Draves***, for example, when a process wishes to access the allocated resource, it simply passes the handle/key pair associated with a shared computer system resource to the kernel. The kernel examines the resource entry indexed by the passed handle to determine whether the passed key is equal to the key in the indexed resource entry. In this way, through the use of handle/key pairs, ***Draves***

provides a system which ensures that only authorized processes are able to access resources. The kernel allows a process access to a resource only when the passed key matches the key for the resource that is stored in the resource entry. See ***Draves,*** column 3, lines 63-67.

To the contrary, claim 1 describes and sets forth requesting to execute at least one of the plurality of instructions or set of instructions by a software code running on the processor and executing the requested instruction or set of instructions providing that the second security ID matches the first security ID. The Examiner, unfortunately, disregards an express teaching of ***Draves*** and removes any distinction between "process" and "resource" terms to make an anticipation rejection. In particular, the Examiner argues that the "process" in ***Draves*** is a "resource." But equating a "process" to a "resource" is inconsistent with ***Draves***, which does not use these terms interchangeably. That is, none of the various resources described by ***Draves*** are either the plurality of instructions or set of instructions or the software code. In other words, none of the various resources described by ***Draves*** include at least one of the plurality of instructions or set of instructions that have been requested to execute by the software code, as set forth in independent claims 1, 9, and 17. As noted above, ***Draves*** describes that a process wishes to access the allocated resource, which is distinct from the process. The Examiner, however, obfuscates this distinction and collapses the two terms into one. Thus, Applicants respectfully submit that claim 1 is not anticipated by ***Draves*** and request that the Examiner's rejections of claim 1 be withdrawn.

For at least the aforementioned reasons, Applicants respectfully submit that the present invention is not anticipated by ***Draves*** and request that the Examiner's rejections of claims 1-3, 7-11, 15-19 and 23-24 under 35 U.S.C. 102(b) be withdrawn.

Claims 4-6, 12-14 and 20-22 stand rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over ***Draves*** in view of ***Krueger*** et al. (U.S. Patent No. 4,962,533). Reconsideration of the present application in view of the reasons set forth herein is respectfully requested.

Applicants submit that claims 4-6, 12-14 and 20-22 are not rendered obvious over ***Draves*** in view of ***Krueger***. To establish a *prima facie* case of obviousness, the prior art reference (or references when combined) must teach or suggest all the claim limitations. *In re Royka*, 490 F.2d 981, 180 U.S.P.Q. 580 (CCPA 1974). The Examiner recognizes that ***Draves*** fails to teach or suggest classifying at least one instruction or set of instructions from a plurality of instructions that are to be executed by a processor as being security sensitive. The Examiner relies upon ***Krueger*** to describe these claim limitations. However, ***Krueger*** does not remedy the fundamental deficiencies of ***Draves*** discussed above.

The cited references also fail to provide any suggestion or motivation for modifying the prior art to arrive at Applicants' claimed invention. To the contrary, ***Krueger*** teaches away from classifying instructions as being security sensitive. For example, in column 2, lines 47-48 and lines 53-56, ***Krueger*** does not check classification of an instruction accessing a word in the memory. Instead, ***Krueger*** is directed to controlling user access to data within a computer system. The computer system classifies data (not an instruction or instructions(s)) only at the level which is needed to provide a security technique for a computer system in which all data retains its classification, and in which no data is overclassified. In a computer system every word in the memory has a corresponding label. This label indicates the security classification, and compartments if any, of that word of data. Each time a word is accessed by any instruction, its classification is checked to see if access is allowed. Any attempt to improperly access any

word within the computer system's memory generates a security violation and prohibits further execution of the currently running process. See ***Krueger***, column 2, 1ines l. 33-56. It is by now well established that teaching away by the prior art constitutes *prima facie* evidence that the claimed invention is not obvious. *See, inter alia, In re Fine,* 5 U.S.P.Q.2d (BNA) 1596, 1599 (Fed. Cir. 1988); *In re Nielson*, 2 U.S.P.Q.2d (BNA) 1525, 1528 (Fed. Cir. 1987); *In re Hedges*, 228 U.S.P.Q. (BNA) 685, 687 (Fed. Cir. 1986).

For at least the aforementioned reasons, Applicants respectfully submit that the present invention is not obvious over the cited references, either alone or in combination. Applicants request that the Examiner's rejections of claims 4-6, 12-14 and 20-22 under 35 U.S.C. 103(a) be withdrawn.

In the Office Action, claims 1-3, 7-11, 15-19 and 23-24 were rejected under 35 U.S.C. § 102(b) as allegedly being anticipated by ***Kamiya*** (U.S. Patent No. 4,949,238). The Examiner's rejections are respectfully traversed.

Kamiya describes an apparatus for detecting memory protection violations in microprogram controlled data processors. To detect a memory protection violation in a data processor for executing microinstructions under control of microprograms, the apparatus comprises privilege level register means for storing a privilege level of a program now being executed. In particular, the data processor comprises a memory protection violation detector 15 and a current privilege level register (CPL) 17 to store the privilege level of a program now being executed are connected. See ***Kamiya***, column 3, 1ines l. 25-27. The memory protection violation detector 15 checks whether the memory protection information stored in the attribute information register 16 is correct or false, on the basis of the memory protection branch microinstruction stored in the mask register 122 of the microinstruction register 12 and the

privilege level value stored in the current privilege level register 17, in order to detect a memory protection violation. See ***Kamiya***, column 3, 1ines l. 35-42. However, ***Kamiya*** is completely silent with regard to requesting to execute at least one instruction by the software code running on the processor and executing the requested instruction. Accordingly, ***Kamiya*** fails to teach or suggest a first security identification (ID) being associated with each of the requested instruction(s) to be executed by a software code with which a second security ID is being associated for restricting the execution of the requested instruction(s) by the software code. ***Kamiya*** also fails to teach or suggest obtaining the second security ID associated with the software code that is requested to execute at least one instruction with which the first security ID is being associated, as set forth in claim 1.

For at least the aforementioned reasons, Applicants respectfully submit that the present invention is not anticipated by ***Kamiya*** and request that the Examiner's rejections of claims 1-3, 7-11, 15-19 and 23-24 under 35 U.S.C. 102(b) be withdrawn.

Claims 4-6, 12-14 and 20-22 were rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over ***Kamiya*** in view of ***Krueger***. The Examiner's rejections are respectfully traversed.

It is respectfully submitted that the pending claims would not have been obvious in view of the prior art of record. To establish a *prima facie* case of obviousness, three basic criteria must be met. First, the prior art reference (or references when combined) must teach or suggest all the claim limitations. *In re Royka*, 490 F.2d 981, 180 U.S.P.Q. 580 (CCPA 1974). Second, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. That is, there must be something in the prior art as a whole to

suggest the desirability, and thus the obviousness, of making the combination. *Panduit Corp. v. Dennison Mfg. Co.*, 810 F.2d 1561 (Fed. Cir. 1986). In fact, the absence of a suggestion to combine is dispositive in an obviousness determination. *Gambro Lundia AB v. Baxter Healthcare Corp.*, 110 F.3d 1573 (Fed. Cir. 1997). The mere fact that the prior art can be combined or modified does not make the resultant combination obvious unless the prior art also suggests the desirability of the combination. *In re Mills*, 916 F.2d 680, 16 U.S.P.Q.2d 1430 (Fed. Cir. 1990); M.P.E.P. § 2143.01. Third, there must be a reasonable expectation of success.

The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on Applicants' disclosure. *In re Vaeck*, 947 F.2d 488, 20 U.S.P.Q.2d 1438 (Fed. Cir. 1991); M.P.E.P. § 2142. A recent Federal Circuit case emphasizes that, in an obviousness situation, the prior art must disclose each and every element of the claimed invention, and that any motivation to combine or modify the prior art must be based upon a suggestion in the prior art. *In re Lee*, 61 U.S.P.Q.2d 143 (Fed. Cir. 2002). Conclusory statements regarding common knowledge and common sense are insufficient to support a finding of obviousness. *Id.* at 1434-35. Moreover, it is the claimed invention, as a whole, that must be considered for purposes of determining obviousness. A mere selection of various bits and pieces of the claimed invention from various sources of prior art does not render a claimed invention obvious, unless there is a suggestion or motivation in the prior art for the claimed invention, when considered as a whole.

As discussed above, ***Kamiya*** fails to teach or suggest a first security identification (ID) being associated with each of the requested instruction(s) to be executed by a software code with which a second security ID is being associated for restricting the execution of the requested instruction(s) by the software code. ***Kamiya*** also fails to teach or suggest obtaining the second

security ID associated with the software code that is requested to execute at least one instruction with which the first security ID is being associated.

The Examiner relies on ***Krueger*** to further describe the first security ID. The Examiner relies upon ***Krueger*** to describe associating a first security ID comprises classifying at least one instruction or set of instructions from a plurality of instructions that are to be executed by a processor as being security sensitive. However, ***Krueger*** is completely silent with regard to classification of an instruction accessing a word in the memory. Instead, to control user access to data within a computer system, the ***Krueger*** computer system classifies data at the level which is needed to provide a security technique. Consequently, ***Krueger*** does not describe or suggest classifying at least one instruction or set of instructions from a plurality of instructions that are to be executed by a processor as being security sensitive.

For at least the aforementioned reasons, Applicants respectfully submit that the Examiner has failed to make a *prima facie* case that the present invention is obvious over the cited references. Applicants request that the Examiner's rejections of claims 4-6, 12-14 and 20-22 under 35 U.S.C. 103(a) be withdrawn.

For the aforementioned reasons, it is respectfully submitted that all claims pending in the present application are in condition for allowance. The Examiner is invited to contact the undersigned at (713) 934-4089 with any questions, comments or suggestions relating to the referenced patent application.

Respectfully submitted,

/Sanjeev K. Singh, Ph.D./
Sanjeev K. Singh, Ph.D.
Rec. No. L0220

AGENT FOR APPLICANTS

Date: November 7, 2006

Williams Morgan & Amerson, P.C.
10333 Richmond Avenue, Suite 1100
Houston, TX 77042
(713) 934-4089 ph
(713) 934-7011 fx

BEFORE THE OFFICE OF ENROLLMENT AND DISCIPLINE
UNITED STATES PATENT AND TRADEMARK OFFICE

LIMITED RECOGNITION UNDER 37 CFR § 11.9(b)

Dr. Sanjeev Kumar Singh is hereby given limited recognition under 37 CFR §11.9(b) as an employee of Williams, Morgan & Amerson, P.C., to prepare and prosecute patent applications for clients of Williams, Morgan & Amerson, P.C. in which a member of Williams, Morgan & Amerson, P.C., is the attorney of record. This limited recognition shall expire on the date appearing below, or when whichever of the following events first occurs prior to the date appearing below: (i) Dr. Sanjeev Kumar Singh ceases to lawfully reside in the United States, (ii) Dr. Sanjeev Kumar Singh's employment with Williams, Morgan & Amerson, P.C. ceases or is terminated, or (iii) Dr. Sanjeev Kumar Singh ceases to remain or reside in the United States on an H-1B visa.

This document constitutes proof of such recognition. The original of this document is on file in the Office of Enrollment and Discipline of the U.S. Patent and Trademark Office.

Limited Recognition No. L0220
Expires: April 14, 2007

Harry I. Moatz
Director of Enrollment and Discipline